**What is Microservices?**

It is a kind of architectural style which structures an application as a collection of Services

Advantage-

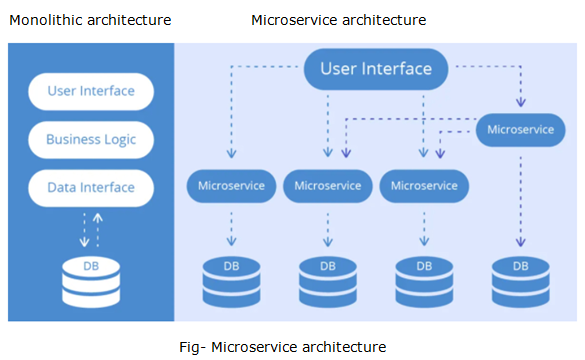
They are loosely coupled.

Independently deployable.

What is the difference between Monolithic application & microservices application?

In Case of Monolithic we consider whole project as a single code-base & hence we deploy the whole app as a single war file.

But when it comes to Microservices application, each of the service is considered as a separate code-base & hence we deploy a separate war file for each code-base (service).



Note

1. We need to create a separate spring boot project for every microservice in the project
2. We need the separate tomcat instances for running each of the application independently.
3. For microservices to communicate with each other , we will use Spring-cloud [EurekaServer].
4. EurekaServer can be considered as a Discovery server.
5. For Microservices to communicate with each other, we have to register them on this EurekaServer.
6. Here each Microservice will be the Client for the EurekaServer.

Example-

It is Kind of site which will display the entire Product that End-User has watched.

Microservice example-

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

@SpringBootApplication

@EnableEurekaClient

public class ProductServiceApplication {

public static void main(String[] args) {

SpringApplication.run(ProductServiceApplication.class, args);

}

}

**package** com.example.demo.controller;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

**public** **class** ProductController {

@GetMapping("/list")

**public** List<String> getRatingList() {

List<String> list = **new** ArrayList<>();

list.add("mobile");

list.add("laptop");

**return** list;

}

}

Application.properties

spring.application.name=productservice

server.port=8083

pom.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.3.4.RELEASE</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.velocity</groupId>

<artifactId>MicroserviceDemos</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>MicroserviceDemos</name>

<description>mrcs project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

<spring-cloud.version>Hoxton.SR9</spring-cloud.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web-services</artifactId>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-netflix-eureka-client -->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

<version>2.2.4.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

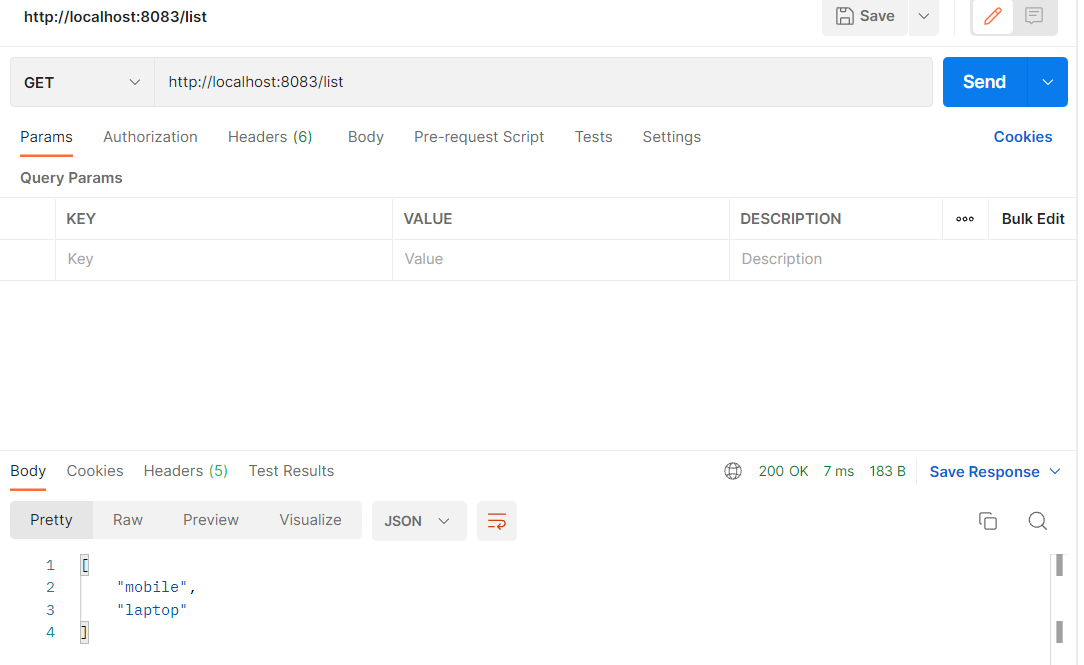
</plugin>

</plugins>

</build>

</project>

Note- test the service using postman whether you will get proper request and response.



How to set up Eureka Server.

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

}

}

Application.properties

server.port = 8761

eureka.client.register-with-eureka = false

eureka.client.fetch-registry= false

pom.xml file

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.3.5.RELEASE</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.velocity</groupId>

<artifactId>eureka-server</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>eureka-server</name>

<description>mrcs project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

<spring-cloud.version>Hoxton.SR9</spring-cloud.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-netflix-eureka-client -->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

<version>2.2.4.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Open the browser and type below url

http//localhost:8761 and hit url and see the microservice will be registered with eureka server.